MiGraNT Project

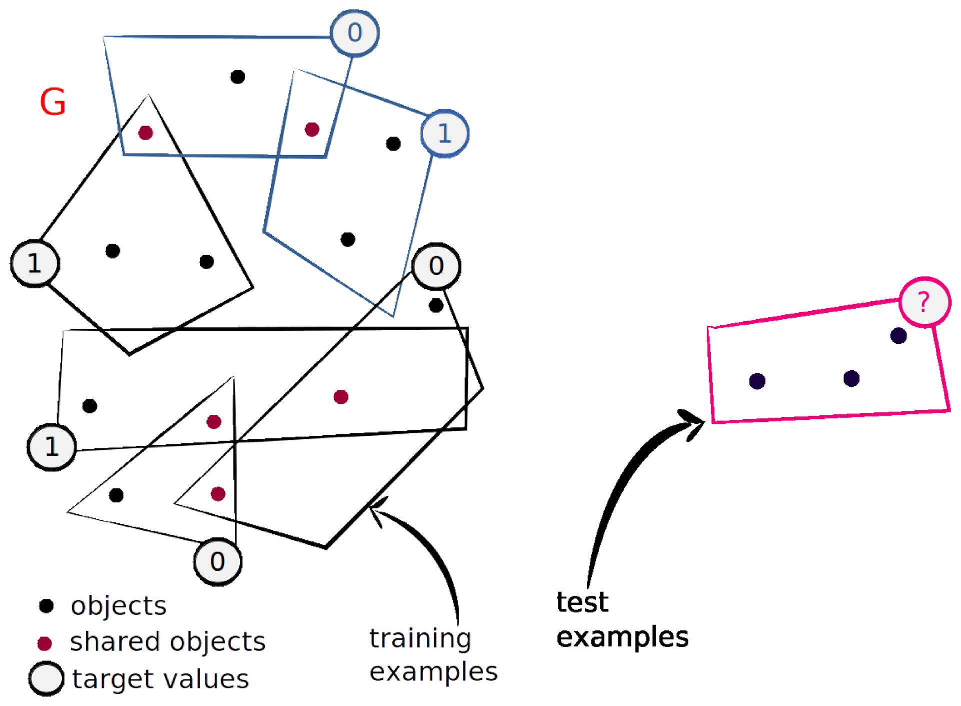
Learning with Networked Examples

Yuyi Wang and Jan Ramon
Department of Computer Science, K.U.Leuven, Belgium

ABSTRACT

An important challenge related to the prediction of relationships between groups of vertices or the properties of such relationships (e.g., link prediction) is that links are not independent (they share vertices) and hence the common assumption that examples are drawn identically and independently does not hold. A related problem occurs in frequent pattern mining [1], where it is non-trivial to define an appealing frequency measure for measuring the support of a pattern in a network. In this paper, we discuss a line of research aiming at solving these two problems in an elegant way by defining a measure which both describes the generalization power of a sample and is an anti-monotonic, normalized graph support measure. We review earlier work, discuss recent results, and suggest directions for future work.

NETWORKED EXAMPLES



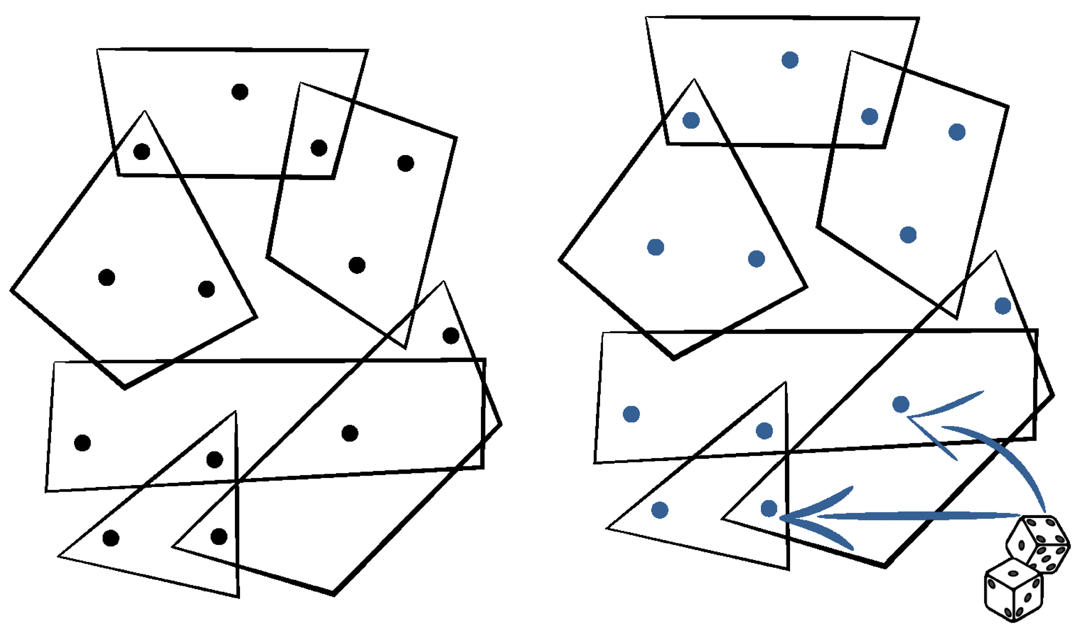
G-networked examples

Every **example** **(hyperedge)** contains several **objects** **(vertices)**.

Objects are assigned **features**.

The **target value** of an example depends on all its objects (their features).

INDEPENDENCE ASSUMPTIONS

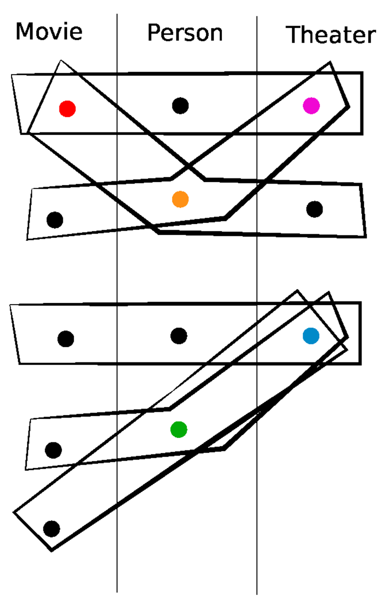


Hyperedges are fixed.

Vertex features are drawn i.i.d..

MOVIE RATING PROBLEM

This is a k-partite hypergraph: vertices can be divided into k partitions and every hyperedge meets every partition exactly once.



	Movie (genre, actor popularity, ...)	Person (age, gender, ...)	Theater (location, ...)
1	movie1	person1	theater1
2	movie1	person2	theater2
3	movie2	person2	theater1
4	movie3	person3	theater3
5	movie4	person4	theater3
6	movie5	person4	theater3

EFFECTIVE SAMPLE SIZE

Given a networked sample S,

PAC: with probability $1-\delta$, the loss is bounded by ε where

$$\delta = \exp(-m(S)\,\varepsilon^2 / C_1 + C_2\varepsilon)$$

with m(S) the **effective sample size**.

The higher m(S) is, the better bound we can expect.

If the examples are independent from each other, then the effective sample size equals to the sample size: $m(S) = |S|$.

EXISTING APPROACHES

Approach 1 EQW (EQual Weight)

We ignore the dependence of the networked examples, and all examples get the same weight.

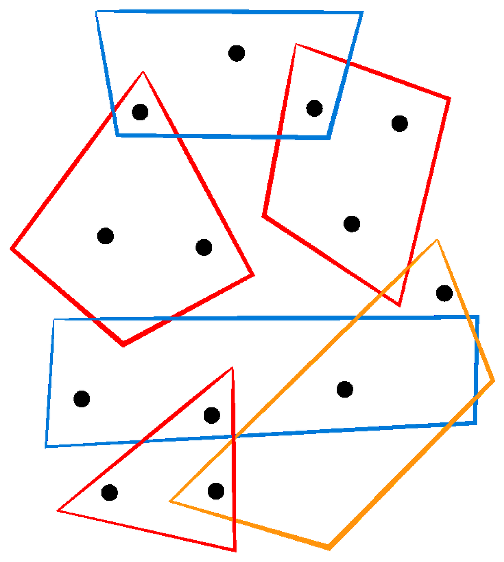
$$m(S) = n_{EQW} = |S| / X'(G)$$

where X'(G) is the fractional edge-chromatic number of the hypergraph G.

A well know result in graph theory is that

$$|S| / X'(G) \le \beta(G)$$

However, it is NP-hard to find a maximum matching in a hypergraph in general.



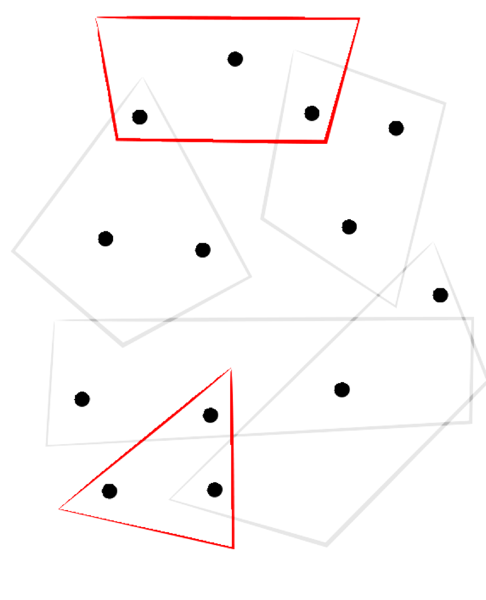
$$n_{EQW} = |S| / X'(G) = 6/3 = 2$$

Approach 2 IND (INDependent set)

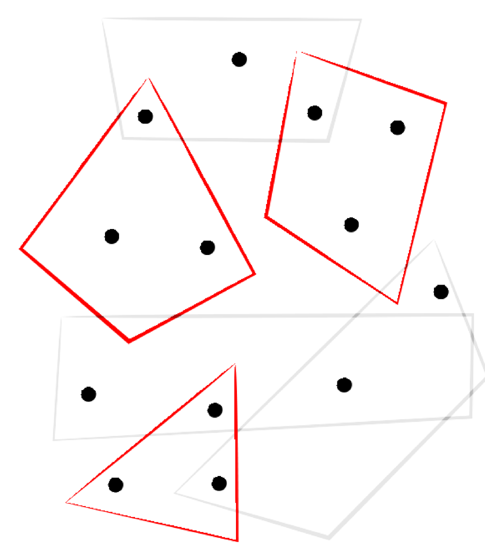
We first choose a subset of training examples S_{IND} (corresponding to a hypergraph matching), and then perform training algorithms on S_{IND}.

$$m(S) = n_{IND} = |S_{IND}| \le \beta(G)$$

where β(G) is the maximum matching size of the hypergraph G.



$$n_{IND} = |S_{IND}| = 2 < \beta(G)$$



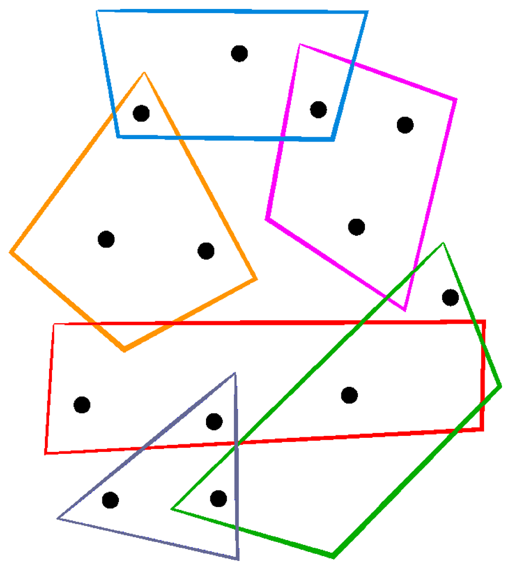
$$n_{IND} = |S_{IND}| = 3 = \beta(G)$$

OUR CONTRIBUTIONS

1. A weighting approach

Every example (hyperedge) has a nonnegative weight.

The sum of the weights of the hyperdges which share a vertex is smaller than or equal to 1.



Maximize $w1+w2+w3+w4+w5+w6$

Subject to

$$w1+w2 \le 1,\quad w2+w3 \le 1,\quad w4+w5 \le 1,\quad w5+w6 \le 1,\quad w4+w6 \le 1,$$
$$w1 \ge 0,\ w2 \ge 0,\ w3 \ge 0,\ w4 \ge 0,\ w5 \ge 0,\ w6 \ge 0.$$

m(S) = s(G) = the sum of the weights and s(G) ≥ β(G).

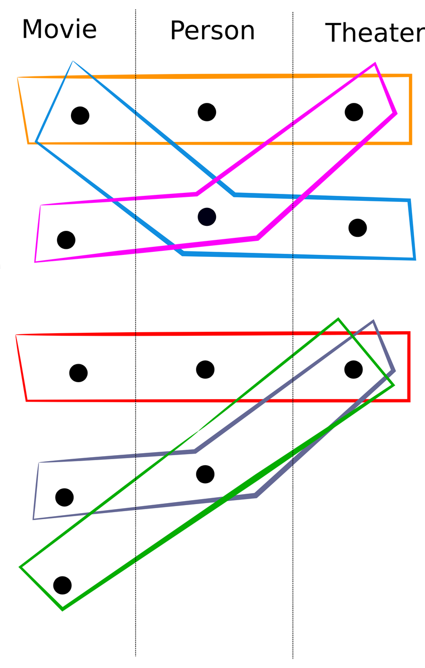
2. Improve m(S) for the EQW method

From $m(S) = |S| / X'(G)$ to $m(S) = \omega(G)$, where ω(G) is the maximum number of hyperedges which share a vertex in G.

$$|S| / X'(G) \le \omega(G)$$

3. The Min-variance strategy

In k-partite case [2], we can also minimize the variance of an estimator (instead of providing an error bound), by weighting the examples.

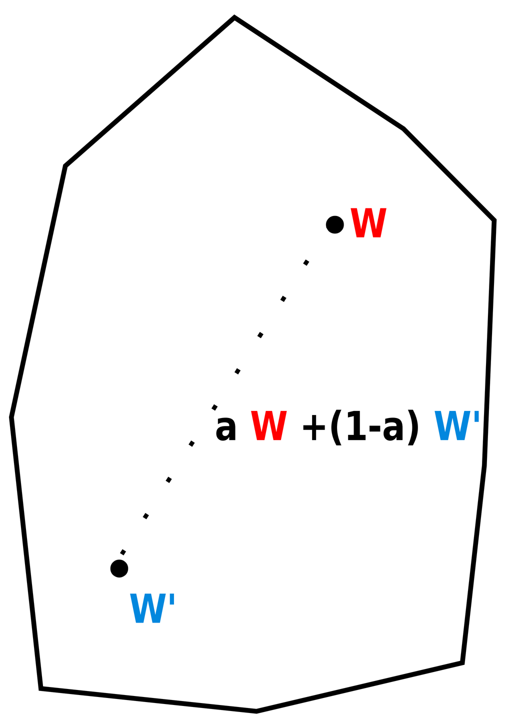


This strategy results in a quadratic program.

4. Convexity and pseudo-convexity

In general, we do not know which weighting vector is optimal.

Still, most concentration bounds (variance, Chernoff, ...) are (pseudo-)convex in the weight vector, and hence if good bounds are available we'll be able to determine good weights.



CONTACT

Yuyi Wang
Email: yuyi.wang@cs.kuleuven.be

Jan Ramon
Email: jan.ramon@cs.kuleuven.be

KU LEUVEN

[1] Wang, Y., Ramon, J., Fannes, T. (2013). An efficiently computable subgraph pattern support measure: Counting independent observations. Data Mining and Knowledge Discovery, 27 (3), 444-477.
[2] Wang, Y., Ramon, J., Guo, Z. (2013). Learning from networked examples in a k-partite graph. Proceedings of the 2013 conference francophone sur l'apprentissage. Conférence sur l'Apprentissage Automatique.
[3] Wang, Y., Ramon, J. (2013). Towards mining and learning with networked examples. Eleventh Workshop on Mining and Learning with Graphs (online Proceedings). Workshop on Mining and Learning with Graphs.